



Genetic algorithms for ambiguous labelling problems

Richard Myers, Edwin R. Hancock*

Department of Computer Science, University of York, York YO1 5DD, UK

Received 15 March 1999

Abstract

Consistent labelling problems frequently have more than one solution. Most work in the field has aimed at disambiguating early in the interpretation process, using only local evidence. This paper starts with a review of the literature on labelling problems and ambiguity. Based on this review, we propose a strategy for simultaneously extracting multiple related solutions to the consistent labelling problem. In a preliminary experimental study, we show that an appropriately modified genetic algorithm is a robust tool for finding multiple solutions to the consistent labelling problem. These solutions are related by common labellings of the most strongly constrained junctions. We have proposed three run-time measures of algorithm performance: the maximum fitness of the genetic algorithm's population, its Shannon entropy, and the total Hamming distance between its distinct members. The results to date indicate that when the Shannon entropy falls below a certain threshold, new solutions are unlikely to emerge and that most of the diversity in the population disappears within the first few generations. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Consistent labelling; Genetic algorithms; Ambiguity; Line labelling; Graph matching

1. Introduction

According to Marr's principle of least commitment, a hallmark of intelligence is the ability to simultaneously entertain several hypotheses until there is sufficient evidence to drop all but one [1]. This paper concerns ambiguous consistent labelling problems, and suggests a framework for maintaining populations of related solutions based on the genetic algorithm.

1.1. Consistent labelling

The consistent labelling problem was formulated by Haralick and Shapiro in the 1970s. A set of units must be assigned labels subject to constraints [2,3]; examples include graph colouring, subgraph isomorphism, inexact matching, the Boolean satisfiability problem and scene labelling. The problem is known to be NP-complete and

is often solved using deterministic search [2,4]. Operators such as forward checking and back marking [4], and Waltz filtering (discrete relaxation) [5], which prune incompatible unit-label assignments from the search space, improve the efficiency of search. However, search is of little use when no totally consistent solution exists, such as is the case with inexact matching or analysis of "impossible" scenes; and neither search nor discrete relaxation use *global* contextual evidence, relying instead on pre-defined local constraint dictionaries.

Most recent work involving consistent labelling has adopted Hummel and Zucker's paradigm for the case where the compatibility coefficients are symmetric: the problem is to find a set of unit-label assignments which maximises some global consistency measure [6]; this is usually done by gradient ascent [6–10]. Gradient ascent techniques are appropriate when there are no local optima between the initial guess and the solution; this is not usually the case, i.e. gradient ascent requires a good initialisation. It may therefore be preferable to use techniques known to possess global convergence properties such as simulated annealing [11,12], mean field annealing [13,14] or genetic search [15], which is the method

*Corresponding author. Tel.: + 1904-433-374; fax: + 1904-432-767

E-mail address: erh@cs.york.ac.uk (E.R. Hancock)

used here. A weakness of global optimisers is that they do not generally take into account the initial labelling assignment. However, it should be said that it is possible to initialise a genetic algorithm in a non-random manner. In this paper, we always use random initialisation.

1.2. Ambiguity

Many consistent labelling problems have more than one possible solution. This was recognised in Waltz's original paper [5], but no strategy for handling ambiguity was developed. In the machine vision literature, ambiguity has been seen as a "bad thing" – to be resolved *locally* as quickly as possible, rather than as a necessary part of scene interpretation. Waltz used search to extract an arbitrary solution [5]; Hummel and Zucker used a simple definition of "unambiguous labelling" as a *sine qua non* for consistency [6]; and Faugeras and Berthod developed a measure of ambiguity which was minimised in their relaxation scheme [7]. Much work concerning ambiguity has been done by linguists and psychologists since language understanding is fraught with ambiguity [16]. MacDonald and coworkers suggest Hummel and Zucker's relaxation framework [6] as a computational model for the disambiguation of sentences based on lexical constraints [16]. Observed frequency and context appear to be the major factors in determining the final interpretation of a word [7,18]; Kawamoto has used a connectionist model to demonstrate this dependency [17].

Ambiguities also occur in visual perception. Connectionist systems have been used to model visual perceptual alternation of ambiguous visual stimuli, in which the interpretation of drawings such as the Necker cube and Schröder staircase (see Fig. 1) periodically switches between several alternatives [19–22]. Bialek and Dewese show that the alternation rate depends on a priori hypotheses [17–22]. Kawabata has observed that the visual fixation point determines the perception of depth and alternation rates in such figures [23]. He suggests that a local interpretation at the fixation point propagates to generate a stable global interpretation. These observations chime with the selective attention hypothesis [24,25] in which a priori expectations combined with focussed attention lead to stable unambiguous interpretations of ambiguous figures. Calliari and Ferrie [26] have recently developed a model-based vision system which can cope with ambiguity. The system makes a set of initial guesses which are refined by subsequent data gathering. This approach has produced promising results, and would seem to complement an active vision strategy. Ambiguity is a major issue in perceptual organisation: Williams and Hanson use an integer linear programming formalism to represent a space of legal labellings from which an optimal one is selected [27]. Kumaran et al. [28] use simulated annealing to find the best of a set of possible organisations of the scene.

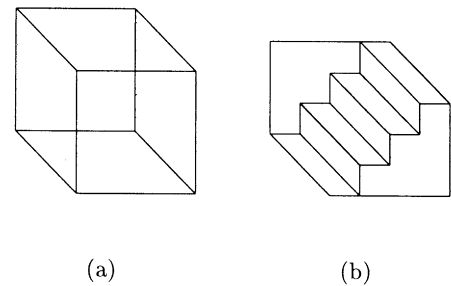


Fig. 1. Two ambiguous drawings: (a) Necker cube, (b) Schröder staircase.

Early disambiguation may be appropriate if there is compelling local evidence for a particular interpretation, but if not, backtracking is generally inefficient [4]. Although the use of *global* contextual information in scene interpretation is a major unsolved problem in machine vision; premature commitment to a particular interpretation does not help – rather, it makes the problem worse. Following the principle of least commitment, the initial stage of scene interpretation should yield several plausible, and perhaps related, solutions from which the system can choose without having to backtrack.

1.3. Paper overview

The aim of the work reported here is to investigate the effectiveness of genetic algorithms as a means of locating multiple solutions to ambiguous labelling problems. Our aim is to explore whether population-based optimisation methods can provide greater solution yields than multiple random starts. We focus on two different labelling problems. The first of these is furnished by Huffman–Clowes line labelling. As we have already pointed out, this is a well-known and extensively studied ambiguous labelling problem. Conventionally, ambiguities are exhaustively generated using the Waltz filtering algorithm. In other words, the line-labelling problem furnishes a convenient example in which the fractional solution yield of the genetic algorithms can be evaluated. However, one of the limitations of the line-labelling problem is the relatively weak nature of the constraints residing in the junction dictionaries for the four line-labels. Our second problem is that of graph matching. Here the dictionaries of consistent subgraph matches provide much stronger constraints. Unfortunately, the exhaustive enumeration of ambiguity is not feasible. We therefore use this second example only to provide additional information concerning the population statistics.

In general, very little modification should be necessary to make a genetic algorithm work for line labelling. An evolutionary process usually has two levels: the *genotypic level* is the level of encodings – chromosomes or

bitstrings; the *phenotypic level* is the level of observed characteristics. In standard formulations, the precise nature of the problem is invisible to the algorithm: cross-over and mutation operate at a genotypic level; selection at the phenotypic. The only stage at which problem knowledge is required is the translation of genotype to phenotype, which is abstracted via a fitness function.

It has become clear that the performance of genetic algorithms can be enhanced by adding a local search step at each iteration [29–31]. Gradient ascent is a standard technique for optimising the consistency of labellings in a relaxation framework [6–9]. Its advantages are speed and simplicity, but it suffers from a major disadvantage in that it cannot escape from local optima. Almost all of the major optimisation techniques which have been developed over the years are attempts to circumvent this problem. Nevertheless, gradient ascent is the method of choice when a local optimum suffices, or when contextual information can provide an initial guess which is close to the global solution.

A combination of the genetic algorithm and gradient ascent should be particularly well-suited to line labelling since the constraints are local. The cross-over used must not be too disruptive because individuals in local optima will tend to have more or less contiguous regions of consistency: a non-disruptive cross-over will cause these regions to coalesce over time. The gradient ascent step will maximise the size of these regions prior to cross-over. Although highly disruptive cross-overs such as uniform cross-over [32] or Eshelman's HUX, in which exactly half of the differing bits are exchanged between the parents [33], explore the search space better, such a cross-over may not be appropriate in a hybrid algorithm because it would undo much of the work of the gradient ascent step. Much exploration of the search space is undertaken in the gradient ascent step: the members of the population will be forced into local optima so the cross-over need not have great exploratory power – indeed, cross-over should be conservative to avoid disturbing the consistent regions. In this framework, the genetic algorithm is seen as an adjunct to gradient ascent rather than the other way around. For this reason, multi-point cross-over should be preferred when gradient ascent is used. However, in non-hybrid genetic algorithms, the need to adequately explore the search space may dictate that a uniform cross-over be chosen [32].

Although the eventual convergence of genetic algorithms using elitist selection is guaranteed [34], it may take arbitrarily long. Some way of ascertaining the current status of the algorithm is needed. The essence of the genetic algorithm is that the cross-over and mutation operators generate diverse solutions which are tested by the selection operator. The notion of “diversity” in a population really incorporates two distinct attributes: the degree of clustering and the extent to which the individuals span the search space.

Our experimental study focusses on two main issues. The first of these is to consider which choice of genetic algorithm gives the best solution yield. There are many algorithm variants reported in the literature. Broadly speaking these can be viewed as deriving from different choices of cross-over and selection operators. Moreover, the different algorithms are governed by the choice of population size and mutation rate. We provide a comparative study which points to the best choice of algorithm and parameter setting to optimal solution yield. The second aspect of our study focusses on the run-time population characteristics. Here our aim is to investigate different population statistics which can be used to monitor solution yield. We consider, three alternatives namely maximum fitness, population entropy and inter-pattern Hamming distance.

The outline of this paper is as follows. Section 2 casts line labelling into an optimisation framework. In Section 3 we explain how the implied optimisation problem can be mapped onto a population-based genetic algorithm. Details of our population-based measures are motivated and presented in Section 4. Section 5 describes experiments for the line-labelling problem. These are augmented in Section 6 with some additional experimentation using graph matching as an example. Finally, Section 7 summarises our conclusions and outlines our future plans.

2. Line labelling by optimisation

Line drawing interpretation has been an active area of investigation in machine vision for over 25 yr: it was the work of Huffman and Clowes on the consistent labelling of line drawings of polyhedral scenes that led Waltz to his seminal discrete relaxation algorithm [5,35,36]. Waltz's contribution was to show how a dictionary of consistent junction labellings could be used in an efficient search for consistent interpretations of polyhedral objects. Such dictionaries are derived from the geometric constraints on the projection of 3D scenes onto 2D planes [5,37]. The interpretation of line drawings remains an important topic in machine vision, and has obvious applications in document analysis, processing architects' sketches, engineering drawings and so-on. Following the work of Huffman, Clowes and Waltz, Sugihara developed a grammar for skeletal polyhedra [37]. Malik has extended the theory to include curved surfaces [38], and Williams has used labelled line drawings to reconstruct smooth objects [39]. Kirousis has developed several efficient algorithms for determining “labellability” and labelling [40]. Most recently, Parodi and Piccioli have developed a method for reconstructing 3D scenes from labelled line drawings given known vanishing points [41].

Hancock and Kittler have built on the work of Faugeras and Berthod [7] and

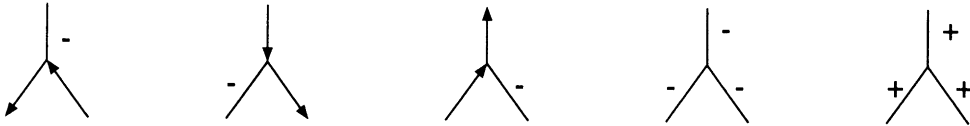


Fig. 2. Legal labellings for a FORK junction.

Hummel and Zucker [6] by developing a Bayesian framework for measuring consistency [8]. This framework can be applied at various levels in image analysis from pixel labelling operations through edge and line labelling to relational matching. Its novelty lies in using an explicit dictionary representation of constraints, as adopted by Waltz, in conjunction with a Bayesian model of the constraint corruption process. The constraint corruption model is based on the premise that the representation of an initially consistent scene is subject to the action of a memoryless label-error process – i.e. a label-corruption process in which successive events are statistically independent [8]. With this model they formulated a probabilistic measurement of the consistency of a labelling: scene interpretation was done by searching for the label configuration which optimised the probability criterion: this was originally done in [8] by gradient ascent.

In a recent preliminary study, Hancock has applied this framework to labelling polyhedral scenes [42]. Suppose that a polyhedral scene under consideration consists of lines drawn from a set $\mathcal{U} = \{u_1, \dots, u_n\}$. Each junction in the scene can be characterised by the set of indices J_k of the lines from which it is constructed. We can form a set $\mathcal{J} = \{J_1, \dots, J_K\}$ whose elements are the tuples of line indices making up each junction.

Each of the ELL, TEE, FORK or ARROW junction types has a distinct dictionary which is a compilation of the permitted label configurations. Suppose that Λ_k denotes the dictionary for the k th junction. If the label set applying to the scene interpretation task is $\Lambda = \{+, -, \rightarrow, \leftarrow\}$, then the cardinality of the junction dictionary $|\Lambda_k|$ is usually much smaller than the number of possible configurations $|\Lambda^{|J_k|}|$. For example, there are only five consistent labellings for a FORK junction (Fig. 2), whereas $4^3 = 64$ combinatorial possibilities exist.

A candidate solution to this labelling problem is a list of labels, $L = \langle \lambda_1, \dots, \lambda_n \rangle$, where $\lambda_i \in \Lambda$. According to Hancock and Kittler’s relaxation framework [8], the global probabilistic criterion is given by summing the probabilities associated with the labellings $L_k \subseteq L$ of each junction, $\Gamma(L_k)$.

$$P(L) = \frac{1}{|\mathcal{J}|} \sum_{k=1}^{|\mathcal{J}|} \Gamma(L_k). \tag{1}$$

The probabilities of the individual junction labellings are computed using a model of the label corruption mecha-

nism. This label-error process assumes that the label on each line is subject to the action of memoryless corruption which occurs with probability p . The consequence of this model is that the consistency of the junction labellings is gauged by an exponential function of their Hamming distances to the various dictionary items. Suppose that $H_{k,l}$ denotes the Hamming distance between the current labelling L_k of the junction $J_k \in \mathcal{J}$ and the dictionary item $l \in \Lambda_k$. The Bayesian model leads to the following expression for the junction probability, Γ .

$$\Gamma(L_k) = \frac{(1-p)^{|J_k|}}{|\Lambda_k|} \sum_{l \in \Lambda_k} \left[\frac{p}{1-p} \right]^{H_{k,l}}. \tag{2}$$

The parameter of this criterion is the probability of memoryless label errors, p . We can re-write the above expression to make the exponential role of Hamming distance explicit.

$$\Gamma(L_k) = \frac{(1-p)^{|J_k|}}{|\Lambda_k|} \sum_{l \in \Lambda_k} \exp \left[-H_{k,l} \ln \frac{1-p}{p} \right]. \tag{3}$$

As the error probability, p , decreases towards zero, labellings lying outside the dictionary make smaller contributions. In the limit of zero label error probability, the global criterion counts the number of consistent junctions. Of greater interest are the observations that for small values of $\ln((1-p)/p)$, the exponential becomes dominated by the term involving the smallest Hamming distance; and that maximising $\sum \exp[-H_{\min}]$ is equivalent to minimising $\sum H_{\min}$ [8]. Thus we can maximise the consistency of a labelling by minimising its *cost*.

$$C(L) = \sum_{k=1}^{|\mathcal{J}|} \min_{l \in \Lambda_k} H_{k,l}. \tag{4}$$

3. Line labelling with a genetic algorithm

Optimisation algorithms based on Darwinian evolution have been proposed by several authors [15,43–45], but it is Holland’s formulation [15] which is regarded as the standard. Genetic algorithms simulate evolution to solve problems: candidate solutions model organisms which exist in an environment modelled by the problem itself. Good solutions to a problem “evolve” over time. The variety of organisms in the world suggests that the problem of survival has many good solutions. It is tempting, therefore, to suppose that a genetic algorithm would

produce several alternative optimal solutions. However, this behaviour has not generally been observed: one solution becomes dominant since selection biases the population in favour of fit individuals. This *genetic drift* can be observed even when survival of individuals is equiprobable. A genetic algorithm could also be suitable for “impossible” objects, where the drawings are not consistently labellable but we nevertheless wish to find one or more of the “next best” labellings.

The algorithm takes a set of bit-strings, the *chromosomes* or *individuals*, and iteratively applies cross-over (mixing) and mutation (random change) operators to them. At every iteration, the fitness of all individuals is evaluated according to some problem-specific measure. Individuals are then selected for the next generation based on their scores. Most implementations terminate when either a specified number of iterations has been performed or a maximally fit individual has emerged. The algorithm has several *control parameters*. These are the cross-over rate, which is the probability of information exchange between individuals; the mutation rate, which in this study is the probability of a single bit-change in an individual; and the population size. The type of cross-over used may also be considered to be a parameter. Where the maximum number of iterations is fixed, this too is a parameter.

Recall from the previous section that a candidate solution to the labelling problem is a list of labels, $L = \langle \lambda_1, \dots, \lambda_n \rangle$, where $\lambda_i \in \Lambda$. If this list is given a binary encoding, $E(L) : L \mapsto I$, where $I \in \{0,1\}^{(n \cdot |\Lambda|)}$, then the problem can be solved using a genetic algorithm, provided some suitable fitness measure $F(I) : I \mapsto [0,1]$ can be derived.

3.1. Fitness measure

We can derive a linear fitness measure directly from the labelling cost in Eq. (4) to turn $C(L)$ into a fitness measure for use in a genetic algorithm (i.e. one with range $[0, 1]$), we exponentiate:

$$F_L(I) = \exp[-\beta C(E^{-1}(I))] \quad (5)$$

This measure falls off rapidly with increasing cost. The steepness of the fall-off can be adjusted by changing the scaling parameter, β (in the work reported here, $\beta = 1$). The function never tolerates more than a few label errors regardless of the number of junctions, for example: F_L has a value of 1 when there are no errors, 0.37 for errors involving one junction, 0.14 for errors involving two junctions, 0.05 for errors involving three junctions, and 0.00 for errors involving six or more junctions.

3.2. Cross-over

Cross-over operators generate two offspring from two parent chromosomes. There are two main classes: uni-

form cross-overs exchange information in a bitwise manner; multi-point cross-overs exchange whole sequences of bits at a time. The cross-over strategy is derived from consideration of the algorithmic variant used, and the relationship between regions in the individual chromosomes and lines in the drawing to be labelled. In a standard genetic algorithm, disruptive cross-overs (i.e. uniform) have been shown to explore the search space better [32,33]. However, in a hybrid genetic algorithm with gradient ascent, much exploration will be accomplished by the gradient ascent step, which will tend to create “islands of consistency”. In this case, a more conservative cross-over (i.e. multi-point), which will preserve and coalesce these islands, should be used.

The use of multi-point cross-over raises the more subtle question of how the structure of the chromosome relates to the structure of the drawing. The cross-over will recombine chunks of chromosome: neighbouring bits will segregate together, a phenomenon known as *linkage* in genetics. It is therefore important that those loci which are close in the chromosome should correspond to lines which occupy the same region of the drawing – i.e. lines which are relatively closely connected. This is not a problem with synthetic data, since humans have a natural tendency to segment line drawings and number junctions and arcs accordingly: thus data can be primed subconsciously to yield solutions. However, the same is not true of real-world data, such as edge-detector output. Our method uses a heuristic to number the arcs. In general, TEE junctions represent occlusions of part of the scene by an overlying plane [35]. A crude segmentation can be achieved by numbering the arcs depth-first, back-tracking at TEE junctions. For our drawings, this makes strongly linked loci in the chromosome map to broadly similar locales in the drawing. However, the inverse relation does not necessarily hold.

4. Monitoring the progress of genetic search

Although the eventual convergence of genetic algorithms using elitist selection is guaranteed [34], it may take arbitrarily long. Some way of ascertaining the current status of the algorithm is needed. The simplest statistics are the maximum and mean fitnesses of individuals. The maximum fitness clearly shows how close the population is to the solution: the mean fitness rapidly approaches the maximum fitness as a result of selection pressure; when a new optimum is found, the mean fitness tends to lag behind the maximum fitness and is not therefore an especially useful statistic.

Probably because of the lack of a coherent, robust theory for genetic algorithms, there has been relatively little effort put in to devising measures of the algorithm's progress at run-time. Many researchers use average fitness to measure the performance (e.g. Ref. [46]). This is

Table 1
Properties of entropy S and average hamming distance, \bar{H}

a	$N_t(x) = 1, N_{t+1}(y) = 1$	S unchanged	\bar{H} unknown
b	$N_t(x) > 1, N_{t+1}(y) = 1$	S increased	\bar{H} unknown
c	$N_t(x) = 1, N_{t+1}(y) > 1$	S decreased	\bar{H} unknown
d	$N_t(x) > 1, N_{t+1}(y) > 1$	increased $N_t(x) > (N_t(y) + 1)$	\bar{H} unknown
		unchanged $N_t(x) = (N_t(y) + 1)$	
		decreased $N_t(x) < (N_t(y) + 1)$	

somewhat naïve since the average fitness will either rapidly approach the maximum fitness as the population converges on an optimum, or provide no specific information if the population is distributed over several local optima. When the positions of the optima are known, the numbers of individuals occupying them or close to them can measure the convergence. However, the positions of optima are usually unknown (or there would not be a problem to solve), and the definition of “close” may entail ungeneralisable assumptions (e.g. Ref. [47]).

Louis and Rawlins use the average Hamming distance between members of the population as a measure of diversity [48]. They successfully use this to give an upper bound on the convergence time of the algorithm, but the measure gives no indication of whether the algorithm is actively exploring the search space or stagnating. Furthermore, as they observed, (traditional) crossover – a key operator in the genetic algorithm – does not affect the average Hamming distance.

The essence of the genetic algorithm is that the crossover and mutation operators generate diverse solutions which are tested by the selection operator. The notion of “diversity” in a population really incorporates two distinct attributes: the degree of clustering and the extent to which the individuals span the search space.

4.1. Clustering

From an information-theoretic point of view, the genetic algorithm’s search space is the alphabet from which a population of symbols is drawn. We wish to obtain information about this space by considering the population. The Shannon entropy is a natural measure of how much information about the space is contained in the population [49], and corresponds to the degree of clustering (a “cluster” is a bag of identical strings in this case).

The Shannon entropy is defined as follows for a bag (population) Ψ of strings, which is a subset of a search space Ξ . Let p_i be the proportion of the i th distinct string in Ψ , such that $\forall i \in [1, |\Xi|] \cdot \frac{1}{|\Psi|} \leq p_i \leq 1$ and $\sum_{i \in \Psi} p_i = 1$.

The Shannon entropy S is given by

$$S = - \sum_1^{|\Psi|} p_i \log p_i. \tag{6}$$

The base of the logarithm depends on the number of possible values of each element in a string. For a standard genetic algorithm this is 2, but since we may not always use a binary encoding, it seems sensible to use the natural logarithm and measure the information in “natural units” [49].

The entropy measures clustering: it is 0 when Ψ contains identical strings; otherwise it is positive and maximal when all the strings in Ψ are distinct, in which case $S = S_{\max} = \log|\Psi|$. Consider replacing some string x with a new string y and the effects of this on the entropy, S , and the average Hamming distance, \bar{H} . There are four cases shown in Table 1 – we use $N_t(x)$ to denote the number of strings x at time t in the population. According to Shannon’s observation that any averaging operation will monotonically increase the entropy [49], if $N_t(x) > (N_t(y) + 1)$, S must increase when an x is replaced by a y .

The entropy monotonically increases as new information is introduced (cases a and b), and monotonically decreases as information is removed (cases a and c). The former behaviour corresponds to exploration of the search space by the genetic algorithm; the latter to convergence of the algorithm. Even when no distinct string has been added or removed, changes in S are predictable. By contrast, \bar{H} is unpredictable in all cases and furthermore tells us nothing about the homogeneity of the population. In fact, \bar{H} is equivalent to $2nq(1 - q)$, where q is the proportion of high bits amongst the distinct strings in the population, and hence says very little about the distribution of the strings themselves.

4.2. Span

As a first approximation, we can measure the extent to which the population spans the search space by considering the total inter-cluster Hamming distance, H_T , which

compares favourably with \bar{H} because it will be increased by any cross-over event which adds new clusters without deleting existing ones. We define H_T by rewriting $\{\Psi\}$ as $\cup \psi_i$, where ψ_i is the i th cluster in Ψ . H_T is given by

$$H_T = \sum_{i=1}^k \sum_{j=i+1}^k H(\psi_i, \psi_j) \quad (7)$$

where k is the number of clusters (distinct strings) in Ψ . H_T will almost certainly be changed by mutation, reflecting the way in which these operators sample the search space.

5. Experiments

The algorithm was tested on three labelling problems with and without gradient ascent and Waltz filtering. Several different parameter sets were tried. The number of iterations required to find a solution and the solution yields were recorded, as were the entropy and total inter-cluster Hamming distance.

We do not give timing data for the algorithm because first such data are generally highly implementation dependent and second our main concern is not algorithm efficiency. Suffice to say that G generations of a genetic algorithm with population size P running on a single processor will require $O(PG)$ cross-overs, mutations and fitness evaluations all of which scale linearly with problem size – we expect the characteristic operation to be fitness evaluation in this case. In the case of the hybrid algorithm, the characteristic operation is definitely fitness evaluation since it involves a quadratic hillclimbing step.

5.1. Method

A generational algorithm was used. The initial population was created at random, and at each generation, all individuals were subject to cross-over and mutation at rates determined by the control parameters. The popula-

tion for successive generations was selected from the current population and its offspring. “Roulette-wheel” selection was used. The algorithm terminated after a set number of iterations regardless of whether any solutions had been found.

The algorithm used was a variant of Eshelman’s *CHC* algorithm [33] in which selection for reproduction is done at random, the parent generation is allowed to mutate, and then parents and offspring compete equally for places in the next generation. “Incest prevention” was not used. HUX cross-over was used in some experiments.

The algorithm was run on the problems shown in Fig. 3. These problems can be made arbitrarily larger by adding disconnected copies; this is reasonable because the algorithm does not “know” that the two drawings are identical: it just sees more lines. The local nature of the constraints means that disconnected copies are almost as difficult as connected copies. In the work reported here, two copies of each drawing had to be labelled. Several parameter sets were tested with and without gradient ascent, and with and without Waltz filtering [5]. Statistics were gathered over sets of 1000 trials.

5.1.1. Control parameters

Control parameters for the genetic algorithm are notoriously difficult to set [50]. The literature recommends two alternative parameter suites as set out in Table 2. These parameters are based on the standard test suite for the genetic algorithm developed by DeJong [51]. Several other sets were tried (Table 3).

5.2. Results

The results are summarised in Tables 4 and 5 (no consistent labellings were found for the impossible object). The algorithm performed best with gradient ascent, and especially well when this was combined with multi-point crossover (Sets *D* and *E*), having the highest convergence rate and highest yields. Waltz filtering

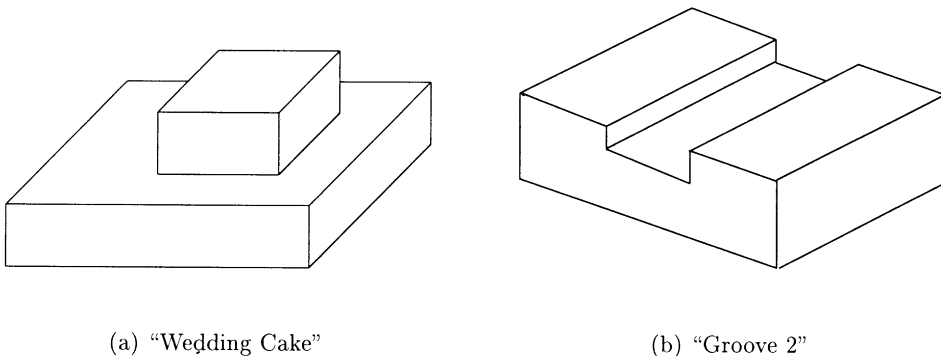


Fig. 3. Test drawings.

Table 2
Parameter sets from the literature

	DJS (DeJong and Spears [52])	Gref (Grefenstette [53])
Population size	100	30
Cross-over type	2 point	Uniform
Cross-over rate	0.6	0.9
Mutation rate	0.001	0.01

Table 3
Additional parameter sets

	Set A	Set B	Set C	Set D	Set E
Population size	100	100	100	100	100
Cross-over type	Uniform	Weighted	HUX	1 point	2 point
Cross-over rate	0.9	0.9	0.9	0.9	0.9
Mutation rate	0.03	0.03	0.03	0.03	0.03

completely confounded the algorithm. The multi-point cross-overs generally outperformed the uniform ones.

5.2.1. Progress measures

Fig. 5 shows sample plots of the maximum fitness, entropy and total inter-cluster Hamming distance of single successful (left column) and unsuccessful (right column) trials. The correlation between the entropy and the total inter-cluster Hamming distance was found to be high (above 0.9) with the gradient ascent hybrid and

lower with the plain algorithm (around 0.7). The correlation between the two measures did not depend on the success of the algorithm. Fig. 6 shows the average population entropy over 1000 trials for plain and hybrid algorithms.

5.3. Discussion

5.3.1. Labelling

The most convincing results were produced when the algorithm was augmented by gradient ascent. All populations converged within five generations on average. This might suggest that the rôle of the genetic algorithm is not significant. However 100,000 restarts of gradient ascent from the same initial conditions only resulted in 84 and 59 consistent labellings for each problem (about 8 and 5%). It is quite clear from this that the hillclimber is getting stuck in local optima, an escape route from which is provided by the genetic algorithm. Yields were highest with multi-point cross-over: this suggests that the algorithm is combining consistent sublabellings, something which uniform cross-overs would impair. The number of generations to convergence (five) compares favourably with the 20 or so needed by the “multiniche crowding” algorithm used by Vemuri [46].

The failure of the algorithm with Waltz filtering may appear surprising: Waltz filtering is known to prune the search space of consistent labellings. However, genetic algorithms work by exploring the fitness landscape; Waltz filtering sharpens this landscape since partially consistent labellings are regarded as being unacceptable. Thus the algorithm is faced with a landscape consisting of several deep troughs, the local minima, from which it cannot readily escape through mutation. The population rapidly converges and no progress can be made.

5.3.2. Similarity of solutions

The solutions found tended to be invariant with respect to FORK junctions. The results of a typical trial

Table 4
Results for the wedding cake problem

	DJS	Gref	Set A	Set B	Set C	Set D	Set E
Standard	<i>c</i> : 2.30% <i>y</i> : 0.06 <i>g</i> : 595	<i>c</i> : 17.8% <i>y</i> : 0.54 <i>g</i> : 528	<i>c</i> : 29.3% <i>y</i> : 2.10 <i>g</i> : 281	<i>c</i> : 30.2% <i>y</i> : 2.27 <i>g</i> : 269	<i>c</i> : 30.4% <i>y</i> : 1.87 <i>g</i> : 305	<i>c</i> : 35.5% <i>y</i> : 3.17 <i>g</i> : 237	<i>c</i> : 38.8% <i>y</i> : 3.45 <i>g</i> : 245
With gradient ascent	<i>c</i> : 99.2% <i>y</i> : 17.0 <i>g</i> : 2.47	<i>c</i> : 76.1% <i>y</i> : 3.34 <i>g</i> : 3.45	<i>c</i> : 99.4% <i>y</i> : 17.3 <i>g</i> : 2.37	<i>c</i> : 97.8% <i>y</i> : 13.5 <i>g</i> : 2.54	<i>c</i> : 99.2% <i>y</i> : 17.6 <i>g</i> : 2.34	<i>c</i> : 100% <i>y</i> : 25.2 <i>g</i> : 2.29	<i>c</i> : 100% <i>y</i> : 33.0 <i>g</i> : 2.22

Note: *c* is the proportion of trials yielding consistent labellings, *y* is the average solution yield over all trials, *g* is the average generation at which the first solutions are found. No solutions were found with Waltz filtering (*c*: 0% in all cases).

Table 5
Results for the groove 2 problem

	DJS	Gref	Set A	Set B	Set C	Set D	Set E
Standard	<i>c</i> : 3.80% \bar{y} : 0.04 \bar{g} : 687	<i>c</i> : 23.3% \bar{y} : 0.34 \bar{g} : 508	<i>c</i> : 38.3% \bar{y} : 1.02 \bar{g} : 230	<i>c</i> : 37.4% \bar{y} : 0.99 \bar{g} : 270	<i>c</i> : 33.3% \bar{y} : 0.80 \bar{g} : 250	<i>c</i> : 42.6% \bar{y} : 1.11 \bar{g} : 244	<i>c</i> : 42.9% \bar{y} : 1.10 \bar{g} : 224
With gradient ascent	<i>c</i> : 98.6% \bar{y} : 9.78 \bar{g} : 2.96	<i>c</i> : 75.9% \bar{y} : 3.23 \bar{g} : 4.23	<i>c</i> : 99.2% \bar{y} : 15.1 \bar{g} : 2.76	<i>c</i> : 99.4% \bar{y} : 13.4 \bar{g} : 2.77	<i>c</i> : 98.4% \bar{y} : 15.3 \bar{g} : 2.77	<i>c</i> : 99.9% \bar{y} : 17.8 \bar{g} : 2.47	<i>c</i> : 99.9% \bar{y} : 19.8 \bar{g} : 2.61

Note: *c* is the proportion of trials yielding consistent labellings, \bar{y} is the average solution yield over all trials, \bar{g} is the average generation at which the first solutions are found.

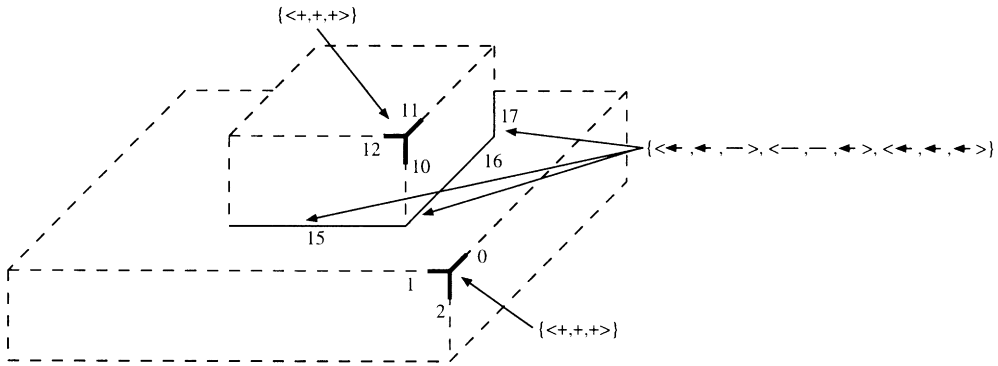


Fig. 4. Related labellings. Labellings of line-triples with strong chromosomal linkage (proximity) found in 11 distinct solutions. Note that the lines incident at FORK junctions only have one label, but the others may have several. Lines are labelled in numerical order.

which found 11 distinct labellings for one of the two “wedding cakes” are given in Fig. 4. The convex interpretation of the two FORKs predominates. This cannot be explained simply by the proximity of the arcs in the drawing (and hence their strong linkage in the chromosomes), since other arc-groups (e.g. 15–17) do not show this consistency.

It is likely that a random change in the labelling of a consistently labelled junction will yield a less good labelling. Consider an ELL junction: there are 16 combinatorial labelling possibilities, six have Hamming distances of zero from the Huffman dictionary (i.e. they are consistent), and ten have Hamming distances of one; none have Hamming distances of two. This means that a random replacement of a consistent labelling has a probability of $5/15 = 0.3$ of yielding another consistent labelling and a probability of $10/15 = 0.6$ of yielding a labelling with a single error. By contrast, a FORK junction has 64 combinatorial possibilities of which five are consistent; the outcomes of a replacement of a consistent labelling are: another consistent labelling with prob-

ability $4/63 = 0.06$, a labelling with Hamming distance one with probability $39/63 = 0.62$, or a labelling with Hamming distance two with probability $20/63 = 0.32$. Thus, the expectation of the Hamming distance from a consistent labelling following a labelling change is 0.6 for an ELL junction and 1.25 for a FORK junction, so FORKs can be said to be more strongly constrained than ELLs. We would therefore expect the labellings of FORK junctions to be relatively immune to the effects of gradient ascent, cross-over and selection; and the final population will probably only contain individuals with one labelling for any particular FORK.

Our results reinforce the findings of Trueswell and others with respect to the propagation of interpretation. Trueswell and coworkers have suggested that rapid disambiguation occurs in regions of strong constraint [54]; Kawabata has suggested that a local interpretation tends to propagate when humans are faced with ambiguous scenes [23]. With this in mind, FORK junctions can be seen as models for strongly constrained localities which tend to dictate the interpretation of their surroundings.

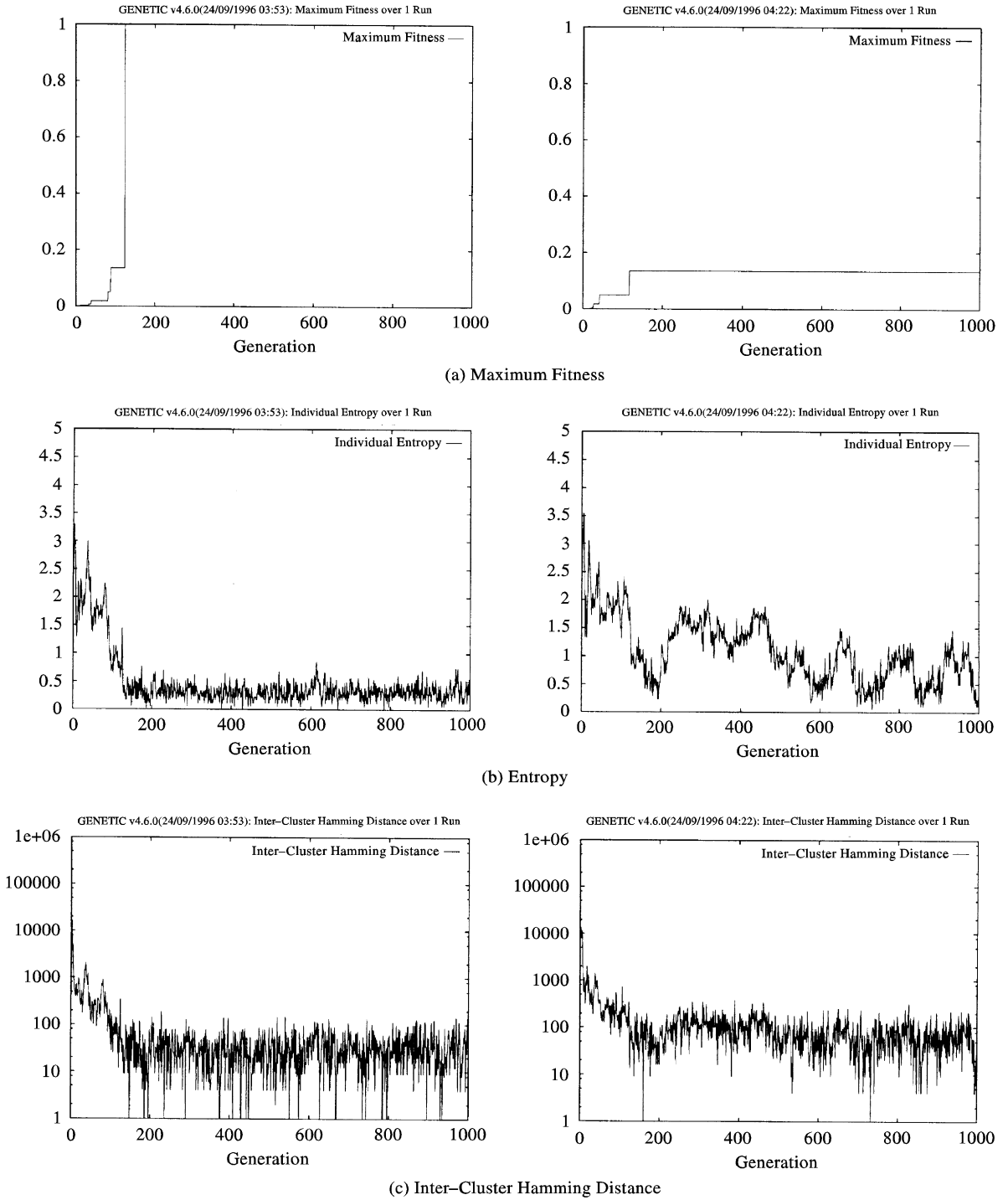


Fig. 5. Measurements on a genetic algorithm. Left column: successful run, right column: unsuccessful run. A log scale is used for the inter-cluster Hamming distance.

This chimes with the notion that the alternative interpretations of a drawing should all be plausible given a priori evidence, and suggests that the search can be controlled by seeding the initial population appropriately.

5.3.3. Progress measures

As can be seen from Fig. 5, for populations of 100 individuals, the entropy always starts at 4.6. This is reassuring: the first generation is initialised at random, and for a population size of 100, the maximum entropy is $\ln 100 = 4.61$. As the population becomes saturated, the entropy usually falls to some minimum below about 2, but the variations in entropy and total inter-cluster Hamming distance after saturation indicate that the algorithm is still attempting to explore the search space. The presence of a set of relatively fit individuals reduces the likelihood that new chromosomes will persist.

Some, but not all, of the major peaks in entropy coincide with jumps in the maximum fitness – i.e. finding a new optimum. Those peaks which do not presumably represent unsuccessful forays in the search space. Those peaks which do coincide with jumps in maximum fitness may either precede or follow them. This can be explained by proposing several methods by which new optimal solutions can arise. The algorithm may explore some fruitful avenue in the search space, causing an increase in entropy, then an optimal solution may be found following a cross-over or mutation. Thus an entropy peak can precede a fitness jump. Alternatively, a new solution may arise *de novo* without extensive search. There will be a fitness jump with no entropy peak. However, if the copy number of the new solution increases over the next few

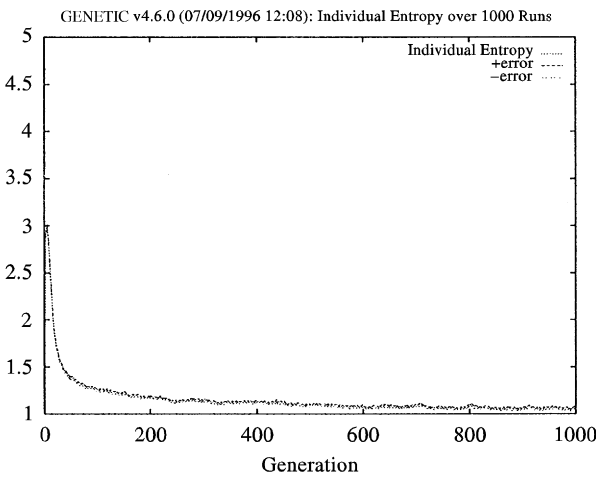
generations, the entropy peak will succeed the fitness jump. A peak occurs because the initial copy number is 1. Replacing a string from a large cluster with one from a smaller one will increase the entropy, but as some point, the cluster containing the new string becomes sufficiently large that adding to it reduces the entropy. Hence the peak.

Fig. 6 shows that the behaviour of the entropy is remarkably consistent between trials: there is an abrupt decrease from the maximum to around $\frac{2}{3}$ of the maximum over the first few generations followed by a fall to some relatively constant minimum value (<2) after 20 to 40 generations. This minimum is typically lower (<1) in successful trials. New optima are rarely found once the entropy minimum has been reached.

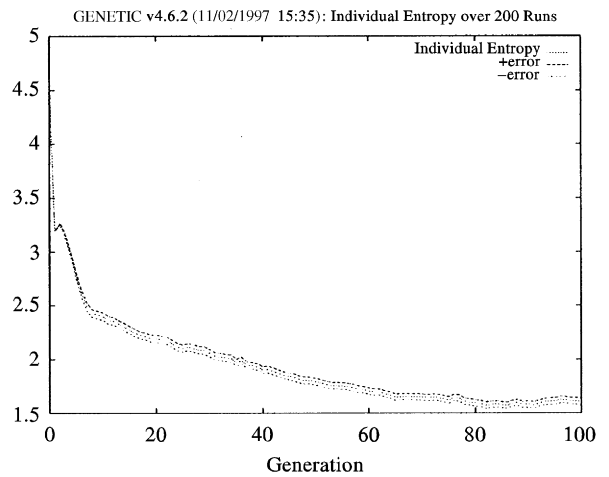
The initial selection removes most of the diversity from the population: the total inter-cluster Hamming distance falls from around 100,000 to around 1000 and the entropy loses $\frac{1}{3}$ of its initial value. This is almost certainly the reason for the high correlations observed between entropy and total inter-cluster Hamming distance. The especially high correlations observed with gradient ascent may arise from the fact that the clusters are relatively stable since they all represent locally optimal solutions.

6. Graph matching

To provide some additional experiments we focus on the problem of graph matching. We furnish this example to illustrate how the performance of the genetic algorithm scales when both the number of available labels and the number of ambiguous solutions increases.



(a) Plain Algorithm



(b) Hybrid Algorithm

Fig. 6. Average entropy of the population for (a) 1000 runs of the plain algorithm and (b) 200 runs of the algorithm with gradient ascent. Lines between points are drawn for clarity: the data are discrete. Dashed lines indicate 1 standard deviation on either side of the solid lines.

Table 6
Algorithm variations used

Experiment	A	B	C	D	E	F	G
Algorithm	GA + GD	GA + GD	GA	CHC + GD	CHC	GA + D	Restarts
Population	100	100	100	50	50	100	100
Iterations	10	10	4000	10	6000	10	10,000
Crossover	2 point	Geomet	2 point	HUX	HUX	None	—
Crossover rate	0.9	0.9	0.9	1.0 ^a	1.0 ^a	0	—
Mutation rate	0.3	0.3	0.3	0.35	0.35	0.3	—

^aIn fact, because of incest prevention, the effective cross-over rate is only about 0.3.

We adopt a simplified version of the inexact matching criterion developed by Wilson and Hancock [10]. In our formulation, we consider only symbolic – i.e. relational constraints: there is no dependence on node attributes. The basic idea underlying this consistency measure is to compare the symbolic matches residing on the neighbourhoods of a data graph with their counterparts in a model graph. Suppose that the data graph $G_1 = (V_1, E_1)$ has node set V_1 and edge set E_1 . In order to accommodate the possibility of clutter nodes in the data graph, we use a null label \emptyset to augment the set of model graph nodes. The basic unit of comparison is the neighbourhood which consists of the nodes connected to a centre object j by data graph edges, i.e. $C_j = j \cup \{i | (i, j) \in E_1\}$. If the model graph is denoted by $G_2 = (V_2, E_2)$, then the state of match between the two graphs is represented by the function $f: V_1 \rightarrow V_2 \cup \emptyset$. The matched realisation of the neighbourhood C_j is represented by the configuration of symbols $\Gamma_j = \cup_{i \in C_j} f(i)$.

Wilson and Hancock’s basic idea was to invoke the concept of a label-error process to facilitate the comparison of the matched neighbourhoods in the data graph with their counterparts in a model graph. This label-error process assumes that mis-assigned matches occur with a probability p while null-matches occur with a probability v . The consequence of this model is that the consistency between the matched data graph neighbourhood Γ_j and the model graph neighbourhood S_k is gauged by two quantities. The first of these is the Hamming distance $H(\Gamma_j, S_k) = \sum_{l \in S_k} (1 - \delta_{f(l), l})$ between the assigned match and the match demanded by the “dictionary-item” S_k . The second quantity is the number of null matches $\Phi(\Gamma_j)$ currently assigned to the nodes of the data-graph neighbourhood C_j . These Hamming distances are used to compute a global probability of match using the following formula:

$$P_G = \frac{1}{|V_1| \times |V_2|} \sum_{j \in V_1} (1 - p)^{C_j} \sum_{k \in V_2} \times \exp[-(\alpha \Phi(\Gamma_j) + \beta_e H(\Gamma_j, S_k))]. \tag{8}$$

The exponential constants appearing in the above expression are related to the uniform probability of matching errors and the null-match probability in the following manner:

$$\beta_e = \ln \frac{1 - p}{p} \tag{9}$$

and

$$\alpha = \ln \frac{(1 - p)(1 - v)}{v}. \tag{10}$$

The parameter p is gradually reduced towards zero with increasing iterative epochs of the genetic algorithm. This has the effect of gradually hardening the constraints residing in the dictionary. In particular, departures from

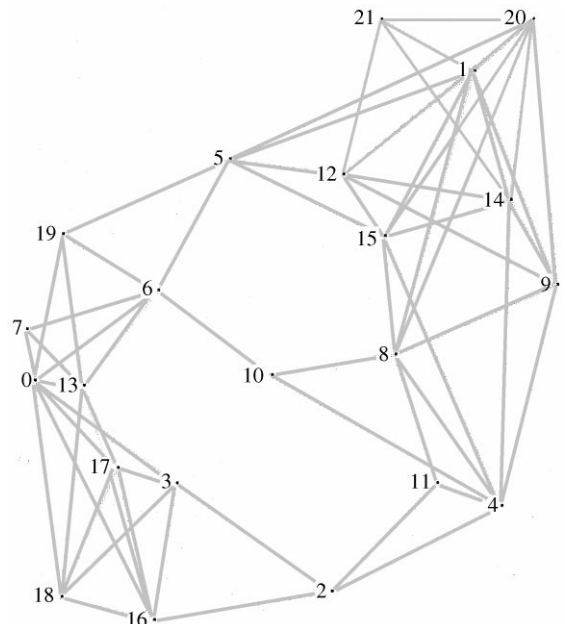


Fig. 7. Data graph. Nodes 20 and 21 are clutter nodes.

Table 7
Results for graph matching 1: $\nu = 0.0001$

Experiment	A	B	C	D	E	F	G
Evaluations per individual	13,400	—	11,200	2380	2670	5080	10,000
Maximum fitness	0.590313	0.590313	0.544882	0.590313	0.590267	0.590313	0.227017
Number of trials	100	100	1	100	5	100	53
Average yield	82.6%	63.1%	1%	48.2%	12%	63%	1%
Mean fitness	= max	= max	0.440464	= max	0.464111	= max	0.205699
Modal Fitness	= max	= max	0.454085	= max	= max	= max	= max

Table 8
Results for graph matching 2: ν automatically assigned (0.095338)

Experiment	A	B	C	D	E	F	G
Evaluations per individual	13,400	—	11,200	3150	2660	5080	10,000
Maximum fitness	0.361458	0.361458	0.339230	0.361458	0.359988	0.361458	0.142249
Number of trials	100	100	1	100	1	100	74
Average yield	81.8%	63.1%	1%	82.8%	14%	62%	1%
Mean fitness	= max	= max	0.273756	= max	0.289400	= max	0.136558
Modal fitness	= max	= max	0.289177	= max	0.314319	= max	= max

zero Hamming distance become increasingly energetically unfavourable. Once $p < \nu$ then residual matching errors migrate into the null category. As we shall demonstrate later, this induces a phase transition which manifests itself as a dip in the different diversity plots.

The quantity P_G lends itself naturally to the definition of a population membership probability. Suppose that $P_G^{(i)}$ denotes the global configurational probability for the i th member of the pool (population) of graphs. By normalising the sum of clique configuration probabilities over the population of matches, we arrive at the following probability for randomly admitting the i th solution to the pool of graphs \mathcal{P} with probability

$$P_s = \frac{P_G^{(i)}}{\sum_{i \in \mathcal{P}} P_G^{(i)}} \tag{11}$$

6.1. Experiments

In our evaluation of the graph matching process, we have focussed on three optimisation algorithms. These were the basic genetic search procedure described in the previous subsection, Eshelman’s *CHC* algorithm [33], and multiple restarts of gradient ascent. In Table 6 we refer to these as “GA”, “CHC” and “Restarts”. We have also investigated the effects of adding a hill-climbing step (“GD” in Table 6) to the genetic search procedure and the *CHC* algorithm. Our experiments have been conducted

with a 20-node synthetic nearest-neighbour (planar) graph to which 10% clutter nodes have been added to simulate the effects of noise, as shown in Fig. 7.

Again, algorithm efficiency is not our primary concern, but we note here that for graph matching, the fitness evaluation is quadratic in the number of nodes, and that the hillclimbing step in this case is quartic.

The algorithm variants and the associated parameters used in our experiments are summarised in Table 6. It must be stressed that no attempt was made to finetune any of the algorithm parameters. In all cases, the probability of null-matches (ν) was set at either 0.0001 (effectively zero) or half the relative cardinalities of the graphs, $(||V_D| - |V_M||)/0.5(|V_D| + |V_M|)$. For genetic algorithms with cross-over, about 10,000 cost function evaluations were allowed. Omitting the cross-over step reduces the number of evaluations required; the other algorithms were run to about 5000 evaluations. Note that the *CHC* algorithm uses half the population of the standard genetic algorithm. Experiment F used no cross-over: it was gradient ascent augmented with mutation and stochastic selection operations.

6.2. Discussion

6.2.1. Matching

The maximal fitness score without null-matches ($\nu = 0.0001$) (see Table 7) is 0.590313 – Fig. 11 shows some sample matches found. The nodes in the lower

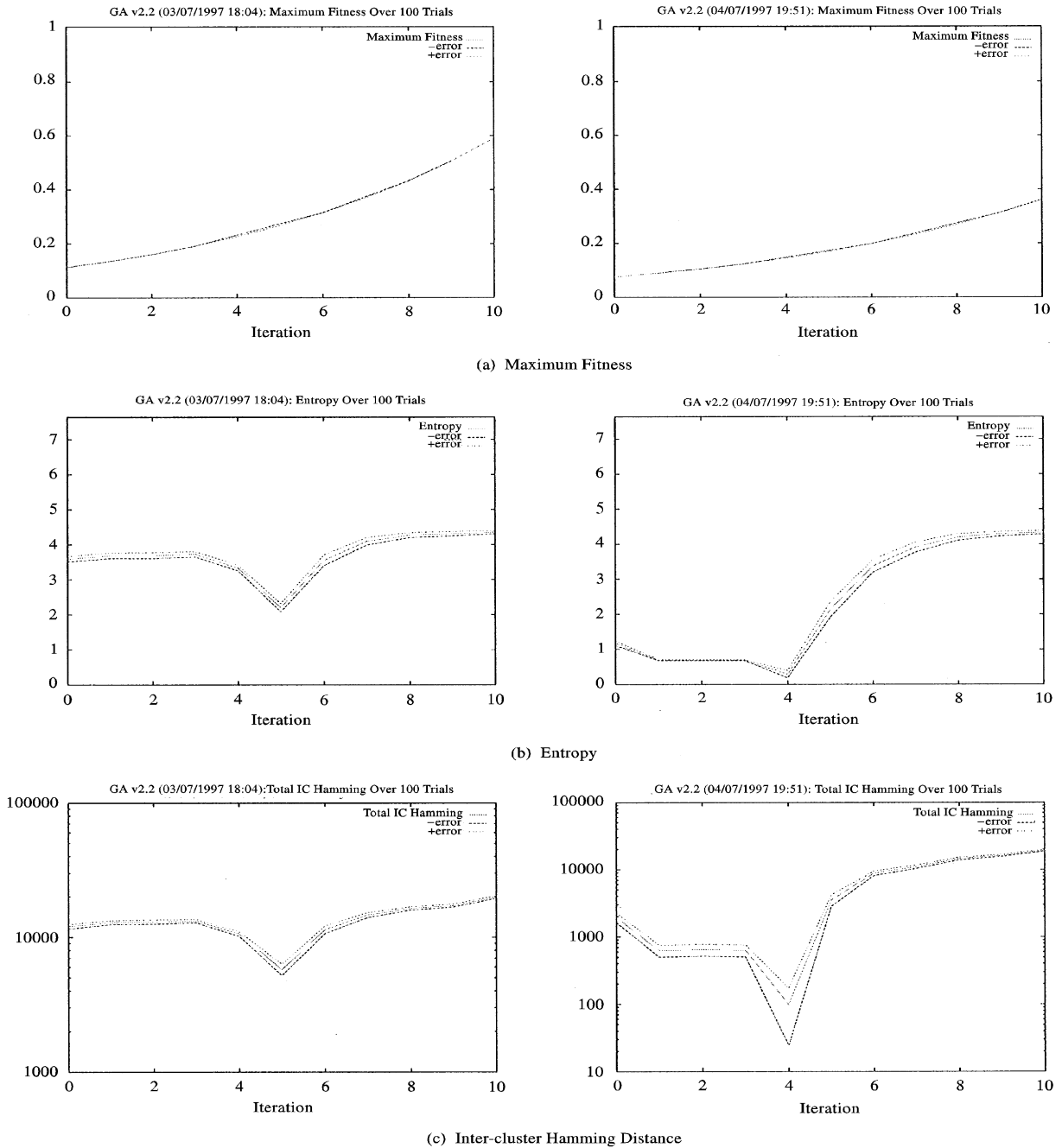
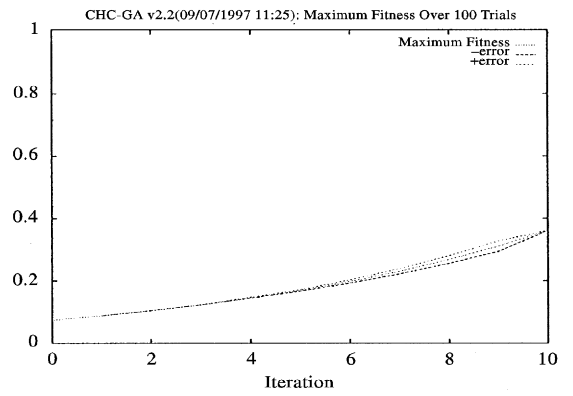
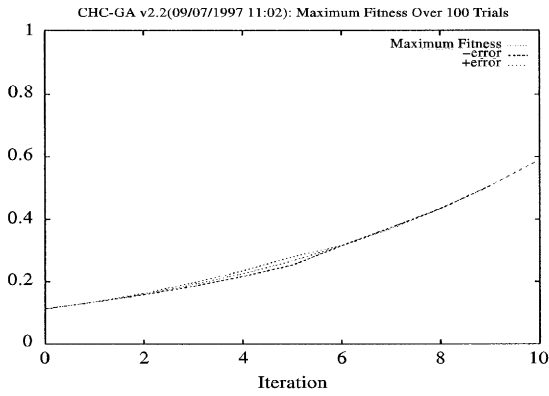


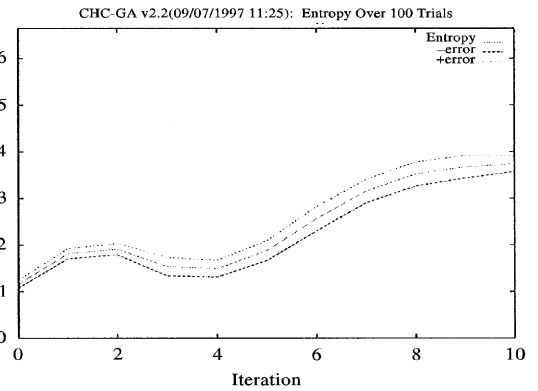
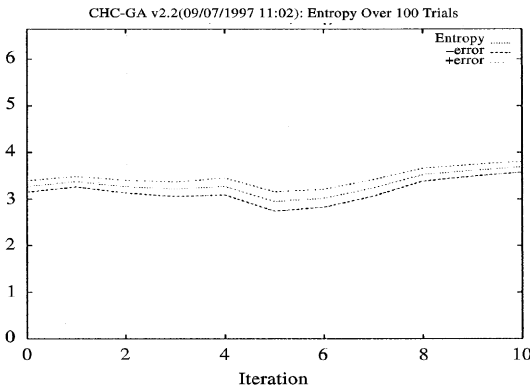
Fig. 8. Measurements on a genetic algorithm. Left column: $v = 0.0001$, right column: $v = 0.095338$. A log scale is used for the inter-cluster Hamming distance.

(uncorrupted) portion of the graph are consistently correctly matched. The nodes which are most often mismatched are 1, 12, 14, 20 and 21. All of these nodes are either clutter nodes or connected by more than 1 clutter edge. Since our cost function only allows 1 null-match

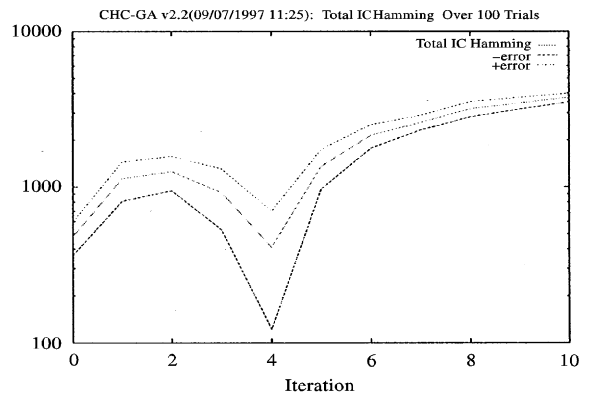
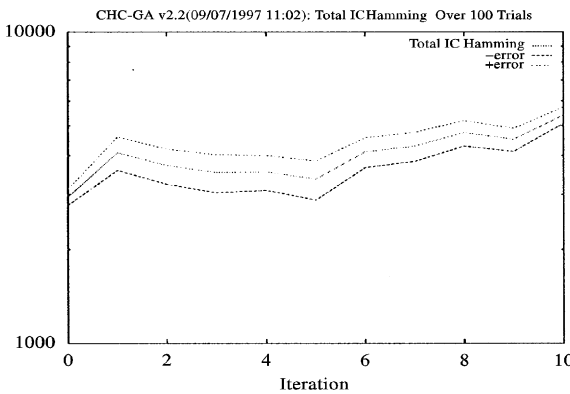
per superclique, it is not surprising that nodes 1, 12 and 14 are mismatched since the cardinality of matching supercliques may not differ by more than 1. Nodes 20 and 21 should be labelled with nulls, but since our cost function discriminates against null-matches, we do not



(a) Maximum Fitness



(b) Entropy



(c) Inter-cluster Hamming Distance

Fig. 9. Measurements on CHC with gradient descent. Left column: $\nu = 0.0001$, right column: $\nu = 0.095338$. A log scale is used for the inter-cluster Hamming distance.

expect the algorithm to get these right. The maximal fitness score with null-matches ($\nu = 0.095338$) (see Table 8) is 0.361458. Although this appears lower than 0.590313, it actually reflects the same number

of errors since the value of ν contributes to the cost function. We may say therefore that the performance of the algorithm on matching is as good as can be expected.

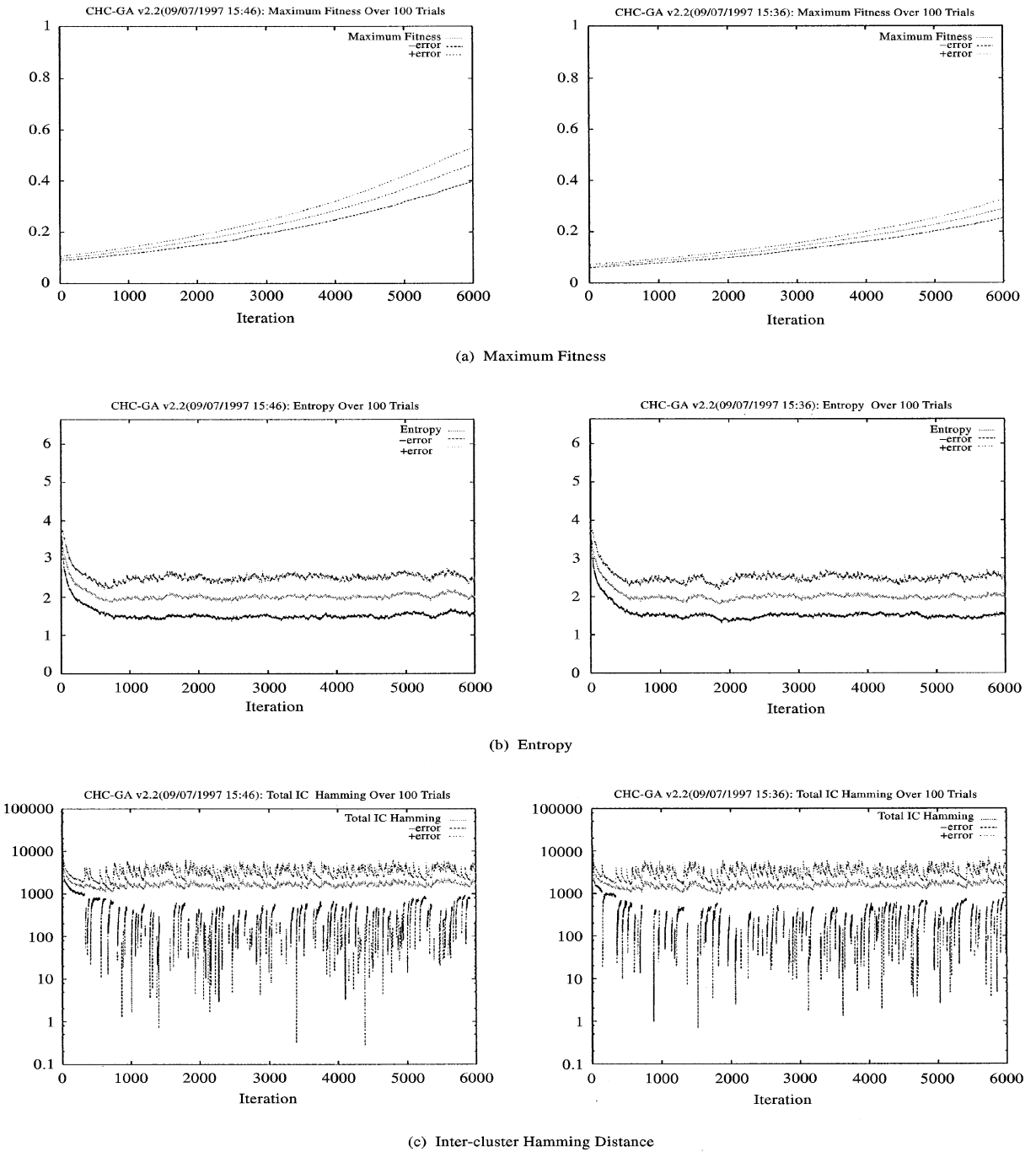


Fig. 10. Measurements on CHC without gradient descent. Left column: $\nu = 0.0001$, right column: $\nu = 0.095338$. A log scale is used for the inter-cluster Hamming distance.

When null-matching was excluded, the genetic algorithm with gradient ascent, CHC with gradient ascent, and “stochastic” gradient ascent gave the best matching results. Results from the genetic algorithm without gradi-

ent ascent were slightly suboptimal. Multiple restarts of gradient ascent did not yield good matches. When null-matching was allowed, all the gradient ascent methods except multiple restarts found optimal solutions. The

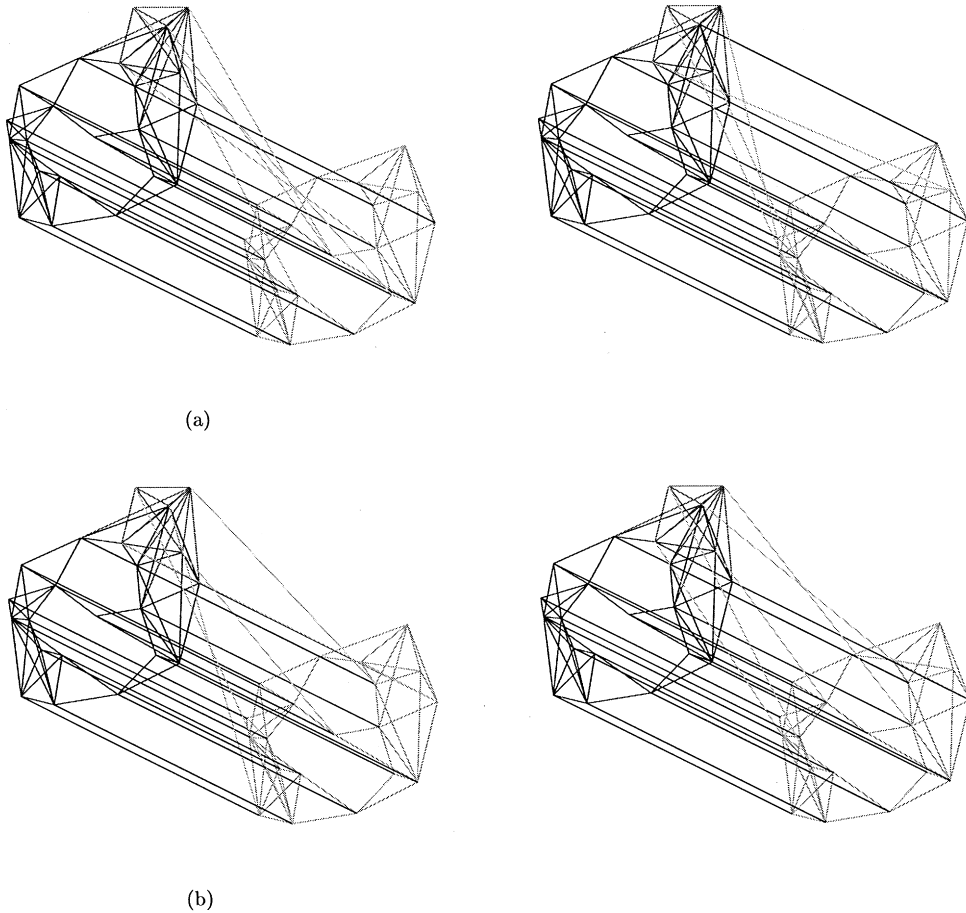


Fig. 11. Typical matches found. This figure shows a random sample of the many solutions of fitness 0.590313.

non-gradient ascent methods performed less well, and multiple restarts performed poorly. Thus, as far as quality of matching is concerned, any of the stochastic optimisers with gradient ascent are adequate for the task.

6.2.2. Ambiguity

The test problem is highly ambiguous: a rough calculation shows that there are tens of thousands of possible solutions. The solution yields from the genetic algorithm with two-point cross-over and gradient ascent were high for both non-null and null matching. Fig. 11 gives typical examples. The yields for the genetic algorithm with geometric cross-over and stochastic gradient ascent were around 20% lower and those of the pure genetic algorithm and multiple restarts very low.

The main conclusion that can be drawn is that the hill-climbing step is important for sustaining population diversity and maintaining ambiguous solutions. This is attributable to the fact that it effectively distributes the solutions in the population to the local optima of

the fitness functions. This has the effect of “pushing” the solutions apart. In this respect it outperforms CHC alone.

6.2.3. Diversity measures

There are several striking differences between the diversity plots for the graph-matching problem shown in Fig. 8 and those already shown for line-labelling in Fig. 5. In the first instance, the fitness measure grows more slowly with iteration number. This feature is attributable to the greater number of labels employed in the case of graph matching. In other words, there are more label swaps to be considered. However, although the process is slower to converge, the population diversity is significantly larger. This is reflected by both the entropy and the inter-cluster Hamming distance. Rather than rapidly decaying, in the case of graph matching both measures are relatively flat, only displaying a mid-epoch dip.

We now make some comments concerning the differences between the diversity measures for the genetic algorithm and the *CHC* algorithm. Fig. 8 shows the

diversity measures for the genetic algorithm with hill-climbing. Figs. 9 and 10 give the diversity measures for the CHC algorithm with and without gradient ascent. The pronounced dip in entropy and total inter-cluster Hamming distance occurs roughly halfway through the algorithm. This is confirmed by other experiments with higher iteration limits. This structure corresponds to a phase transition induced by the onset of the condition $p < v$ as the error-probability is annealed with iteration number. It is only at this point that null-labels enter the match.

These plots confirm the conclusion that gradient ascent sustains diversity better than CHC. Combining gradient ascent with CHC results in further improvements.

7. Conclusion

Consistent labelling problems frequently have more than one solution. In order that global contextual information be brought to bear in image analysis, several interpretations of locally ambiguous regions should be maintained. We have argued that most work in the field has aimed at disambiguating such regions early in the interpretation process, using only local evidence.

Our primary contribution has been to show that the genetic algorithm are a robust tool for solving the line-labelling problem and hence other consistent labelling problems. When combined with gradient ascent and using a multi-point crossover, the algorithm robustly finds multiple solutions to the problem. These solutions are related by common labellings of FORK junctions, which are the most strongly constrained of all junction types considered. The number of generations to convergence of the algorithm compares very favourably with that reported for multi-niche crowding, which also finds several solutions [46]. These conclusions are reinforced by the graph-matching study.

There is no solid theory to predict the behaviour of genetic algorithms or suggest appropriate parameter values. As a result, most of the run-time performance measures found in the literature are naïve. We have proposed three run-time performance measures: the maximum fitness of the population, the Shannon entropy of the population, and the total Hamming distance between distinct clusters of individuals. The maximum fitness and Shannon entropy provide useful information about the status of the algorithm. The total inter-cluster Hamming distance appears to be highly correlated with the Shannon entropy, especially with the gradient ascent hybrid. The results to date indicate that a population with a Shannon entropy of less than 2 has become saturated, and that new solutions are unlikely to emerge from such a population for some considerable time. Furthermore, most of the diversity in the population disappears in the first few iterations.

References

- [1] D. Marr, Vision, Freeman, New York, 1982.
- [2] R.M. Haralick, L.G. Shapiro, The consistent labelling problem: Part 1, IEEE Pattern Anal. Mach. Intell. 1 (1979) 173–184.
- [3] R.M. Haralick, L.G. Shapiro, The consistent labelling problem: Part 2, IEEE Pattern Anal. Mach. Intell. 2 (1980) 193–203.
- [4] R.M. Haralick, G.L. Elliott, Increasing search tree efficiency for constraint satisfaction problems, Proc. 6th Int. Joint Conf. on Art. Intell., 1979, 356–364.
- [5] D. Waltz, Understanding line drawings of scenes with shadows, in: P.H. Winston (Ed.), Psychology of Computer Vision, McGraw-Hill, New York, 1975, pp. 19–91.
- [6] R.A. Hummel, S.W. Zucker, On the foundations of relaxation labeling processes, IEEE Pattern Anal. Mach. Intell. 5 (1983) 267–287.
- [7] O.D. Faugeras, M. Berthod, Improving consistency and reducing ambiguity in stochastic labeling: An optimisation approach, IEEE Pattern Anal. Mach. Intell. 3 (1981) 412–424.
- [8] E.R. Hancock, J. Kittler, Discrete relaxation, Pattern Recognition 23 (1990) 711–733.
- [9] R.C. Wilson, E.R. Hancock, Graph matching by discrete relaxation, in: E.S. Gelsema, L.N. Kanal (Eds.), Pattern Recognition in Practice, 4, Elsevier, Amsterdam, 1994, pp. 165–176.
- [10] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, IEEE Pattern Anal. Mach. Intell. 19 (1997) 634–648.
- [11] C.D. Gelatt, S. Kirkpatrick, M.P. Vecchi, Optimisation by simulated annealing, Science 220 (1983) 671–680.
- [12] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Pattern Anal. Mach. Intell. 6 (1984) 721–741.
- [13] D. Geiger, F. Girosi, Parallel and deterministic algorithms from MRFs: surface reconstruction, IEEE Pattern Anal. Mach. Intell. 13 (1991) 401–412.
- [14] A.L. Yuille, J.J. Kosowsky, Statistical physics algorithms that converge, Neural Comput. 6 (1994) 341–356.
- [15] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MA, 1975.
- [16] N.J. Pearlmutter, M.C. MacDonald, M.S. Seidenberg, The lexical nature of syntactic ambiguity resolution, Psychol. Rev. 101 (1994) 676–703.
- [17] A.H. Kawamoto, Nonlinear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account, J. Memory Language 32 (1993) 474–516.
- [18] N.J. Pearlmutter, M.C. MacDonald, Individual differences and probabilistic constraints in syntactic ambiguity resolution, J. Memory Language 34 (1995) 521–542.
- [19] J.A. Feldman, D.H. Ballard, Connectionist models and their properties, Cognitive Sci. 6 (1982) 205–254.
- [20] M. Riani, F. Masulli, E. Simonotto, Neural network models of perceptual alternation of ambiguous patterns, in: S. Levialdi, V. Cantoni, L.P. Cordella, G. Sanniti di Baja (Eds.), Progress in Image Analysis, World Scientific, Singapore, 1990, pp. 751–758.
- [21] M. Riani, E. Simonotto, Stochastic resonance in the perceptual interpretation of ambiguous figures – a neural network model, Phys. Rev. Lett. 72 (1994) 3120–3123.

- [22] W. Bialek, M. Deweese, Random switching and optimal processing in the perception of ambiguous signals, *Phys. Rev. Lett.* 74 (1995) 3077–3080.
- [23] N. Kawabata, Visual fixation points and depth perception, *Vision Res.* 18 (1978) 853–854.
- [24] N. Kawabata, T. Mori, Disambiguating ambiguous figures by a model of selective attention, *Biol. Cybernet.* 67 (1992) 417–425.
- [25] K.L. Horlitz, A. O’Leary, Satiation or availability – effects of attention, memory and imagery on the perception of ambiguous figures, *Perception Psychophys.* 53 (1993) 668–681.
- [26] F.G. Callari, F.P. Ferrie, Active recognition: using uncertainty to reduce ambiguity. Proceedings of the 13th International Conference on Pattern Recognition, 1996, pp. 925–929.
- [27] L.R. Williams, A.R. Hanson, Perceptual completion of occluded surfaces, *Comput. Vision and Image Understanding* 64 (1996) 1–20.
- [28] K. Kumaran, D. Geiger, L. Parida, Visual organisation for figure/ground separation, *CVPR 1996*, pp. 155–160.
- [29] L.S. Davis (Ed.), *A Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
- [30] C. Graves, D. Whitley, R. Beveridge, K. Mathias, Test driving three 1995 genetic algorithms: new test functions and geometric matching, *J. Heuristics* 1 (1995) 77–104.
- [31] R.C. Wilson, A.D.J. Cross, E.R. Hancock, Genetic search for structural matching, in: B. Buxton, R. Cipolla (Eds.), *Proceedings of the Fourth European Conference on Computer Vision*, vol. 1, 1996, pp. 514–525.
- [32] G. Syswerda. Uniform crossover in genetic algorithms. in: *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, pp. 2–9.
- [33] L.J. Eshelman, The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. in: G.J.E. Rawlins (Ed.), *Foundations of Genetic Algorithms*, vol. 1, Morgan Kaufmann, Los Altos, CA, 1991, pp. 265–283.
- [34] G. Rudolph, Convergence analysis of canonical genetic algorithms, *IEEE Trans. Neural Networks* 5 (1994) 96–101.
- [35] D.A. Huffman, Impossible objects as nonsense sentences, in: B. Meltzer, D. Michie (Eds.), *Machine Intelligence*, vol. 6, Edinburgh University Press, 1971, pp. 295–323.
- [36] M.B. Clowes, On seeing things, *Artificial Intelligence* 2 (1971) 79–116.
- [37] K. Sugihara, Picture language for skeletal polyhedra, *Comput. Graphics Image Process.* 8 (1978) 382–405.
- [38] J. Malik, Interpreting line drawings of curved objects, *Int. J. Comput. Vision* 1 (1987) 73–103.
- [39] L.R. Williams, Topological reconstruction of a smooth manifold-solid from its occluding contour, in *ECCV 92*, 1992, pp. 36–47.
- [40] L.M. Kirousis, Effectively labeling planar projections of polyhedra, *IEEE Pattern Anal. Mach. Intell.* 12 (1990) 123–130.
- [41] P. Parodi, G. Piccoli, 3D shape reconstruction by using vanishing points, *IEEE Pattern Anal. Mach. Intell.* 18 (1996) pp. 211–217.
- [42] E.R. Hancock, An optimisation approach to line labelling, in: S. Impedovo (Ed.), *Progress in Image Analysis and Processing*, vol. 3, World Scientific, Singapore, 1994, pp. 159–165.
- [43] A.S. Fraser, Simulation of genetic systems by automatic digital computers, *Austral. J. Biol. Sci.* 10 (1957) 484–491.
- [44] H.J. Bremermann, The evolution of intelligence. The nervous system as a model of its environment, Technical report, Department of Mathematics, University of Washington, Contact No. 477(17), 1958.
- [45] R. Toombs, J. Reed, N.A. Barricelli, Simulation of biological evolution and machine learning, *J. Theoret. Biol.* 17 (1967) 319–342.
- [46] V.R. Vemuri, W. Cedeño, T. Slezak, Multiniche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments, *Evolutionary Comput.* 2 (1995) 321–345.
- [47] D.R. Bull, D. Beasley, R.R. Martin, A sequential niche technique for multimodal function optimisation, *Evolutionary Comput.* 1 (1993) 101–125.
- [48] S.J. Louis, G.J.E. Rawlins, Syntactic analysis of convergence in genetic algorithms, in: D. Whitley (Ed.), *Foundations of Genetic Algorithms*, vol. 2, Morgan Kaufmann, Los Altos, CA, 1993, pp. 141–151.
- [49] C.E. Shannon, A mathematical theory of communication, *Bell System Tech. J.* 27 (1948) pp. 379–423.
- [50] L.J. Eshelman, J.D. Schaffer, R.A. Caruna, R. Das, A study of control parameters affecting online performance of genetic algorithms for function optimisation, in: *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, pp. 51–60.
- [51] K.A. DeJong, An analysis of the behaviour of a class of genetic adaptive systems, Ph.D. Thesis, University of Michigan, Department of Computer and Communication Sciences, 1975.
- [52] K.A. DeJong, W.M. Spears, An analysis of the interacting rôles of population size and crossover in genetic algorithms, in: *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, Springer, Berlin, 1990.
- [53] J.J. Grefenstette, Optimisation of control parameters for genetic algorithms, *IEEE SMC* 16 (1986) 122–128.
- [54] M.K. Tanenhaus, J.C. Trueswell, S.M. Garnsey, Semantic influences on parsing: use of thematic rôle information in syntactic disambiguation, *J. Memory Language* 33 (1994) 285–318.

About the Author—RICHARD MYERS took his B.A. in Natural Sciences from the University of Cambridge in 1989. In 1995 he gained an M.Sc. with distinction in Information Processing at the University of York. He is currently working towards a D.Phil. in the Computer Vision Group at the Department of Computer Science at the University of York. The main topic of his research is the use of genetic algorithms to solve consistent labelling problems arising in the machine vision domain. In 1997 he spent two months working at NEC Corporation in Kawasaki, Japan, sponsored a REES/JISTEC fellowship. His interests include evolutionary computation, perceptual organisation and labelling problems.

About the Author—EDWIN HANCOCK gained his B.Sc. in physics in 1977 and Ph.D. in high energy nuclear physics in 1981, both from the University of Durham, UK. After a period of postdoctoral research working on charm-photo-production experiments at the Stanford Linear Accelerator Centre, he moved into the fields of computer vision and pattern recognition in 1985. Between 1981 and 1991, he held posts at the Rutherford-Appleton Laboratory, the Open University and the University of Surrey. Dr. Hancock is currently Reader in the Department of Computer Science at the University of York. He leads a group of some 15 researchers in the areas of computer vision and pattern recognition. He has published about 180 refereed papers in the fields of high energy nuclear physics, computer vision, image processing and pattern recognition. He was awarded the 1990 Pattern Recognition Society Medal and received an honorable mention in 1997. Dr. Hancock serves as an Associate Editor of the journal *Pattern Recognition* and has been a guest editor for the *Image and Vision Computing Journal*. He is currently guest-editing a special edition of the *Pattern Recognition* journal devoted to energy minimisation methods in computer vision and pattern recognition. He chaired the 1994 British Machine Vision Conference and has been a programme committee member for several national and international conferences.